Prediction Of Transport Protocol Performance Through Simulation

K. Mills, M. Wheatley, and S. Heatley
National Bureau of Standards
Gaithersburg, Maryland 20899

ABSTRACT

A five-layer simulation model of OSI protocols is described and applied to predict transport performance on a local area network (LAN). Emphasis is placed on time-critical applications typical of a small, flexible manufacturing system. The results predict that, with current technology, OSI protocols can provide 1.5 Mbps throughput, one-way delays between 6 and 10 ms, and response times between 15 and 25 ms. The results also indicate that CSMA/CD is a reasonable access method for time-critical applications on small, factory LANS, if loads of less than 40% are anticipated. For loads between 40% and 70%, a token passing access method provides better performance for time-critical applications.

I. INTRODUCTION

To bridge the automation and information islands found in today's factories and offices, a significant portion of United States industry has selected the international standard open systems interconnection (OSI) protocols as described in the General Motor's manufacturing automation protocols (MAP) specification [MAP85] and the Boeing Computer Services' technical and office protocols (TOP) specification [TOP85]. The source for both MAP and TOP is the OSI reference model [OSI82] and the series of protocol and service specifications that followed. Demonstrations at the 1984 National Computer Conference and the 1985 Automated Factory exhibition have shown that OSI protocols can provide effective interworking over multiple vendor networks and internetworks; however, questions remain concerning the efficiency of the protocols for planned applications and subnetwork technologies. What range of throughputs can be expected with present technology? What delays and response times can be achieved in factory and office applications? Are new protocol mechanisms required to meet performance needs? Must the seven layer architecture be modified for special time-critical applic - ations? If so, what modifications are required? These questions and others are the subject of protocol performance research being conducted at the National Bureau of Standards (NBS).

This paper reports some early performance predictions concerning the OSI transport protocol, class 4, in use for several simple factory applications. The basis for these predictions are results obtained from a simulation model constructed to aid development of a plan for live performance experiments [HEA85A]. First, a concise description of the simulation model is given. Then, simulation results are presented for file transfer, periodic status reporting, and request-response applications. For each application, transport performance is evaluated over both IEEE 802.3 and 802.4 networks [IEE84A, IEE84B, IEE85].

II. The Model

The simulation model is an enhanced version of a model previously used to predict class 4 transport protocol performance for bulk data transfer over satellite channels [MIL85, COL85A, COL85B]. Only significant enhancements are described; the references provide a more detailed understanding of the model.

The model permits simulation of two or more detailed transport stations communicating over a point-to-point link, a carrier sense multiple access with collision detection (CSMA/CD) local network, or a token passing bus. If a point-to-point link is simulated, each direction is treated as a simplex channel so that a variety of technical arrangements can be modeled. For example, a point-to-point link may be routed over a satellite hop in one direction while the return channel is carried over a fiber optic cable. If a local network bus is simulated, the medium is shared between the transport stations and, optionally, some number of background stations. Since the results reported in this paper are derived from modeling local area network (LAN) architectures, emphasis in the discussion is placed on LAN models.

1986 ACM 0-89791-201-2/86/0800-0075

One goal of the NBS performance research is to eliminate the traffic pattern assumptions normally used in analytical and simulation modeling. Such modeling usually restricts application traffic to a set of arrival rate and message size distributions. With such scenarios, the arrival of user messages at a station is independent of the arrival of user messages at any other station and is also independent of the arrival of any previous messages at the station. This independence assumption does not represent real behavior for most applications. To eliminate this assumption, each transport station includes one or more application models that exhibit a specific pattern of behavior.

The applications modeled include: simplex and full-duplex file transfer, status reporting, request-response transactions, database query, data entry, and virtual terminal. Each application denotes a special pattern of message exchanges and the experimenter may provide the arrival process and size distribution for each type of message in the exchange. The experimenter may also assign a probability to each element of a set of alternate message sizes. This is convenient to represent interactive applications where some messages are short while others are long.

Due to the increased range of applications modeled, measures of one-way delay and response time, as seen by the transport user, have been added to the set of metrics. Other metrics have been added to allow the experimenter to evaluate the effects of network conditions on overall user performance. Most transport metrics are provided on a per connection basis enabling evaluation of the relative performance provided to each user. As an aid in identifying bottlenecks, resource utilization metrics are available. The resources are the CPU, memory, and the network channel. The transport process is also represented as a resource to show the allocation of transport CPU time to transport connections.

A. Architecture of the Simulated Local Network

The system modeled is a small, flexible manufacturing system (FMS) with a set of programmable controllers connected along a coaxial cable bus. Figure 1 illustrates the specific network architecture to be discussed. The network is small, covering one thousand feet and comprising twelve stations. The experiments, to be discussed, are limited to performance as seen at a pair of transport stations operating on a network with ten additional stations. The purpose of the transport stations is to provide detailed simulation of five protocol layers, including one or more manufacturing applications (APPL), the class 4 transport protocol (TP), a type one, class one logical link control protocol

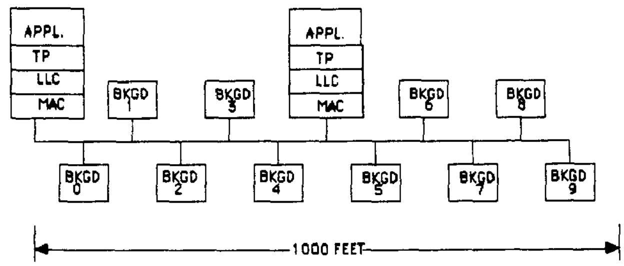

FIGURE 1. SIMULATED LOCAL AREA NETWORK LAYOUT

(LLC), a media access control protocol (MAC), and a physical protocol. The additional stations are used to generate a network background load.

The specific networks simulated are a baseband coaxial bus with a CSMA/CD access method and a broadband coaxial bus with a token passing access method. The simulated characteristics for the CSMA/CD and token bus networks are shown in Tables 1 and 2, respectively. CSMA/CD simulation is accomplished with a model developed by Sharon Heatley of the NBS. The Heatley model provides accuracy commensurate with very detailed CSMA/CD simulations while achieving execution efficiency that permits integration of the detailed transport station and the CSMA/CD network models [HEA85B].

TABLE 1.
SIMULATED CSMA/CD CHARACTERISTICS

Link Speed	10 Mbps Propagation
Delay	400 ns
Slot Time	51.2 us
Jam Time	3.2 us
Interpacket Delay	9.6 us
Preamble	7 octets
Minimum Packet	64 octets
Maximum Packet	1518 octets
MAC Header	19 octets
LLC Header	3 octets
Network Header	1 octet

Simulation of the token bus is similar in concept to a model developed at the NBS by Jean-Luc Archambault [ARC84]. The token bus model, excluding such functions as logical ring management and recovery from lost token, is integrated into the detailed transport station model to permit the simulation of network background load, and to provide realistic modeling of the token access method, including all optional priority classes as specified in the IEEE 802.4 standard. The token bus model is efficient and can be executed with the transport station model in a reasonable amount of CPU time.

All of the simulated subnetwork technologies (CSMA/CD, token bus, and point-to-point links) have access to an error generation function, simulating a range of bit error

TABLE 2.
SIMULATED TOKEN BUS CHARACTERISTICS

Link Speed	10 Mbps
Propagation Delay	800 ns
Head End Latency	10 us
P6 Token Holding Time	410 us
P4 Target Rotation Time	10 ms
P2 Target Rotation Time	10 ms
P0 Target Rotation Time	10 ms
First Preamble	4 octets
Interframe Preamble	2 octets
Token Size	19 octets
Token Processing Time	50 us
Minimum Packet	23 octets
Maximum Packet	8018 octets
MAC Header	19 octets
LLC Header	3 octets
Network Header	1 octet

rates. For the experiments discussed in this paper, the error generation function is disabled, thus the networks are simulated with error-free transmission. This is not unreasonable when typical LAN error rates of 1 bit in 10**8 or better are assumed, as required by the IEEE 802.3 specification [IEE84B].

B. Architecture of the Simulated Transport Stations

The results in this paper assume a hardware and software architecture for the simulated transport stations that is consistent with present technology and typical of several available OSI protocol products. Figure 2 provides a block diagram of the hardware architecture assumed at each transport station and shows the mapping of protocol functions to hardware blocks. Three autonomous CPU chips provide parallel operation of applications, transport and link control, and media access. Such an architecture is typical for minicomputer and microcomputer

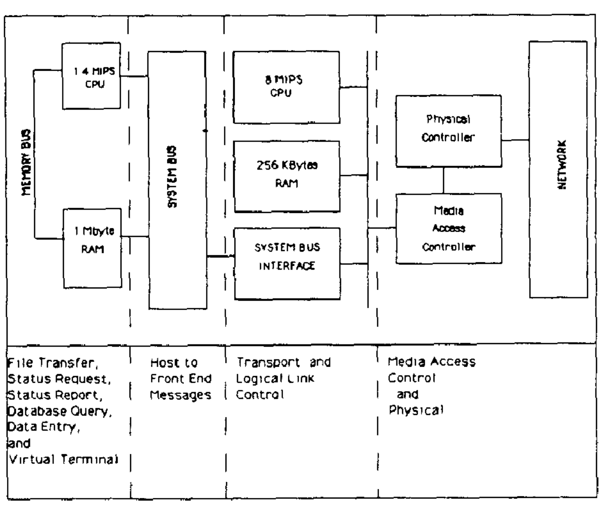

Figure 2 Simulated Foreground Station - Hardware and Software

systems that are implemented as a set of boards connected through a system bus. In the simulation, one set of boards makes up the host processor, shown in Figure 2 as a 1.4 MIPS CPU and 1 Mbytes of random access memory (RAM).

The access from host CPU to host memory is over a private bus, thus keeping the system bus available for host to peripheral communications. The communication subsystem is implemented as a single-board, front-end processor comprising a .8 MIPS CPU chip, 256 Kbytes of local memory, a system bus interface, a local bus for on-board communication, and a media access controller chip with a physical interface. The data communication applications reside on the host while the transport and lower layer communications software reside on the front-end board. Communication between the host and front-end board is provided via shared access to host memory across the system bus. A simple, confirmed message passing protocol is assumed for relaying host to front-end messages through the shared memory area.

Media access control is modeled as a separate co-processing chip that operates in conjunction with the communications CPU. Communication between the co-processors is provided by interrupts and a shared area in the local memory. This is a realistic approach to implementing layered protocols for both the CSMA/CD and token passing access methods. Co-processor chips have been announced or are available for both networks. Products with an architecture similar to that modeled by the NBS are available or announced for a variety of layered protocol suites [MCL85, RYD85, HAB85, INT84, NRC84].

In the following sections, simulation results for several application scenarios are reported in the form of graphs. Some of the discussion relies on the key abbreviations shown in Table 3.

TABLE 3. KEY ABBREVIATIONS

TSDU	Transport Service Data Unit A user message.
IDU	Interface Data Unit TSDUs are divided into IDUs for passing between host and front-end.
TPDU	Transport Protocol Data Unit A message with a transport header.
DT	Data TPDU
AK	Acknowledgement TPDU

III. File Transfer Performance Predictions

Within a flexible manufacturing system (FMS), device reprogramming is accomplished by downloading a new program to each controller. This type of operation can be modeled as a file transfer. The experiments described in following paragraphs are designed to predict transport station throughput for one-way (i.e., single connection) and two-way (i.e., two connection) file transfers over both a CSMA/CD network and a token bus. For each experiment, the background load on the local network is varied between 0 and 70% (0-7 Mbps).

A. One-Way File Transfer

The one-way file transfer experiment models a single connection between two transport stations with 1 Mbyte of data flowing from a source user to a destination user, transport layer acknowledgements flowing in the opposite direction. The user messages are continuously queued and are segmented into 1485 byte TPDUs. Each TPDU is placed in a separate link layer packet. The simulated background load is increased from 0 to 70% in 5% increments. Background messages, generated by a Poisson arrival process, are either 32 or 576 bytes with a probability of .75 and .25, respectively. These message sizes are characteristic of measured LAN traffic at Xerox PARC [SHO80]. The load is controlled by varying the average arrival rate. The results are presented in the graph shown as Figure 3.

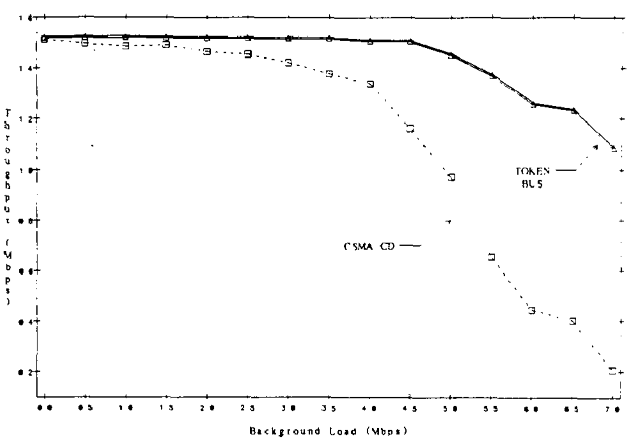

Figure 3. Transport User Throughput for a One-Way File Transfer

Predicted transport throughputs under no load are about 1.5 Mbps for both the CSMA/CD network and the token bus. As the load increases to 15% and beyond, the transport throughput falls for the CSMA/CD network while remaining more stable for the token bus. Beyond 40% load, transport throughput on the CSMA/CD net falls steeply. On the token bus, station throughput remains stable up to 50% load and then falls gradually until it reaches 1.1 Mbps at 70% load. At 70% load, predicted transport throughput on the contention bus is a scant 206 Kbps.

Previous performance studies for CSMA/CD networks by Shoch and Hupp [SHO80] at Xerox PARC and Toense at the NBS [TOE83] show that the medium throughput is stable (i.e., it will reach some maximum and remain there as the load increases) and the access method is fair (i.e., the share of throughput seen by each station will be approximately equal). In the present study, at 70% load, the CSMA/CD medium throughput is 7.3 Mbps while the transport user transmitting the file sees .206 Mbps throughput. In this experiment, the protocol efficiency, which is the ratio of user bits transmitted to total bits transmitted including protocol headers and transport retransmissions (not including collision fragments and preamble), is .52. This means that the transmitting transport station sees a throughput of .396 Mbps at the MAC layer. This is about 5% of the 7.3 Mbps total medium throughput. There are twelve stations and each should receive 8.3% of the medium throughput. The transport station is thus getting less than a "fair share" of the medium. In the studies cited above, all the stations were offering the same load to the network. In the experiments described here the transport station and the background stations are offering different loads to the network.

One factor contributing to decreased transport station throughput is error recovery overhead. At high background loads, some TPDUs are involved in repeated collisions and are discared by the MAC protocol. Discarded TPDUs are recovered only after retransmission timer expiration within the transport layer. Both the actual TPDU retransmission and the idle time due to error recovery result in lowered transport station throughput.

A second factor contributing to the lowered transport throughput is a substantial increase in the average medium access delay for the two transport stations. Medium access delay is the difference between the time a packet is ready for transmission and the time that the first bit is successfully transmitted. Medium access delay includes time for deferring and backing off. As the background load increases, the average and standard deviation of the medium access delay for the transport station begins to diverge from the average and standard deviation of the media access time measured for all stations on the networks. The divergence begins at 50% load where the average medium access delay for the network is 1.73 ms, while the average medium access delay for the transport stations is 3.53 ms. At 70% load the network average is 1.78 ms, while the transport station average is about 24 ms.

The difference in average and standard deviation of medium access delay results from a combination of: 1) the inherent properties of the CSMA/CD backoff and 2) modeling artifact. When a packet collides, the CSMA/CD access method requires the random selection of a backoff time from a uniform distribution that grows exponentially in range as a function of the number of collisions for the packet. Thus, a packet with N collisions enjoys an advantage when colliding with a packet with N + 1 or more collisions because the range of the backoff distribution is smaller. Under heavy network loads, access to the medium occurs in bursts for a given station because, once a station's packet has won the contention, subsequent packets from the same station have an advantage over the remainder of the contending stations. The practical effect of this procedure under heavy loads is that, after acquiring the medium, the station retains the medium until all packets queued for transmission have been set.

The modeling artifact, interacting with the bursty nature of the medium acquisition at heavy load, is the difference in transmission queue size between the background and transport stations. Background stations are simulated to have infinite length transmit queues while the transport station's simulated send queues hold four packets. Thus, the maximum burst from a transport station is four packets, while the maximum burst observed for background stations was forty packets.

The throughputs achievable with a token bus are much greater than on a CSMA/CD network because the maximum packet size permitted by the IEEE 802.4 standard is larger. Increased throughput is possible when the fixed overhead of message processing is the largest part of the total message processing burden. In the experiments reported here, no transport checksumming is used, and thus, the processing that depends upon message size is a single memory-to-memory copy from a user buffer to a network transmit buffer. Figure 4 shows throughput results for a 1 Mbyte file transfer using 5120 byte TPDUs. The

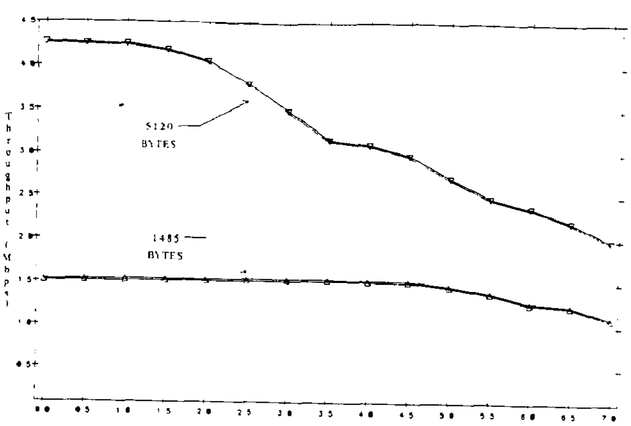

Figure 4. Transport User Throughput for Two TPDU Sizes on a Token Bus

transport throughput has increased 280% over that obtained in the previous experiment (i.e., with 1485 byte TPDUs). However, throughput for the larger TPDU size is much more sensitive to the background load; throughput being only 50% greater at 70% load. This result is due to a longer token rotation time when the transport station TPDUs are larger, 21.5 ms average versus 9.3 ms average. Since transmission of the large TPDU takes longer, 4 ms versus 1.2 ms, more arrivals occur at the background stations, leading to increases in token rotation time. The standard deviation in token rotation time also increases, ranging from 4.6 to 4.9 ms for 5120 byte TPDUs versus 2.2 to 2.4 ms for 1485 byte TPDUs.

B. Two-way File Transfer

The two-way file transfer experiment models two connections between the same pair of transport stations with 512 Kbytes flowing per connection. Each transport station is the source for one connection and the destination for another. The background traffic is the same as in the one-way file transfer experiment.

The results are shown in Figure 5. The general trend in throughput is the same as for the previous experiment, with the token bus allowing more stable performance and the CSMA/CD network exhibiting decreases in transport throughput starting at 15% load and falling sharply past a load of 40%. Two

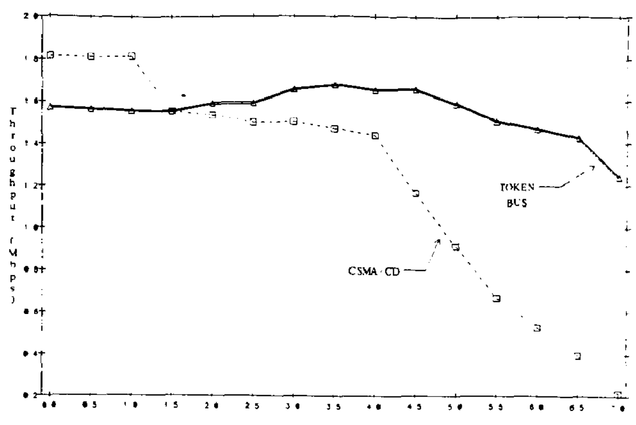

Figure 5. Transport User Throughput for a Two-Way File Transfer

specific differences are evident for the two-way file transfer as compared with the one-way. First, the curves are higher because some previous idle time on the part of the transport CPU that was formerly only a receiver has been put to use. Second, the transport throughput obtained on an unloaded CSMA/CD network is 224 Kbps higher than the transport throughput on the token bus. This result is because the CSMA/CD access method imposes no medium access delay on an unloaded network. Therefore, the transport station CPUs can achieve utilizations of 1. (Note that in the one-way file transfer case, activity within the transport stations is unbalanced; the sender and receiver achieve CPU utilizations of 1 and .67, respectively. In the two-way case, activity is balanced and both CPUs are fully used). The token bus will always impose some medium access latency because the transport station must wait for the token to rotate before transmitting any messages. For the experiments reported here, the average token rotation time on the unloaded token bus was 3 ms. Due to this latency, transport station CPUs achieved utilizations of only .87, leading one to expect 13% less throughput on the unloaded token bus; exactly the result obtained.

IV. Status Reporting Performance Predictions

A common factory application is the periodic reporting of status between programmable controllers. In such applications, the primary performance concern is one-way delay. The experiment reported in the following paragraphs evaluates the expected one-way delay for a 20 millisecond periodic status reporting application between two transport stations. The local network background traffic varies from none to 70% in 5% increments and comprises 200 byte messages generated by a Poisson arrival process. One-way delay is measured as shown in Figure 6. A TSDU of 200 bytes is given to the sending transport at time Ts, encapsulated as a TPDU, and sent across the network to the receiving transport, arriving at the receiving user at time Te. The one-way delay (Te-Ts) is computed and accumulated for later calculation of the minimum, maximum, average, and standard deviation of the one-way delays measured on

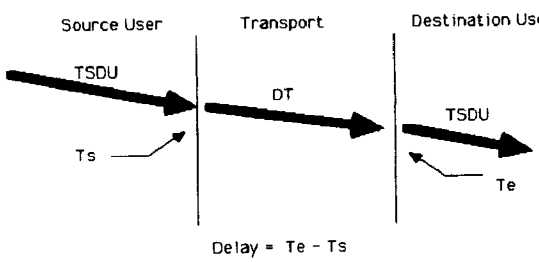

FIGURE 6. Measurement of One-Way Delay

the transport connection. For each sample, 500 status report trials are used. The results are given in Figures 7 and 8.

The expected one-way delays for transport users over the CSMA/CD net are lower than the delays over the token bus until a cross over point is reached, at about 60% load, after which average one-way delays on the CSMA/CD net increase steeply. The user delays seen across the token bus, although somewhat higher for loads below 60%, increase in a more gradual manner. Therefore, if anticipated network loads are below 60%, the CSMA/CD local network provides better average one-way delay for the transport user.

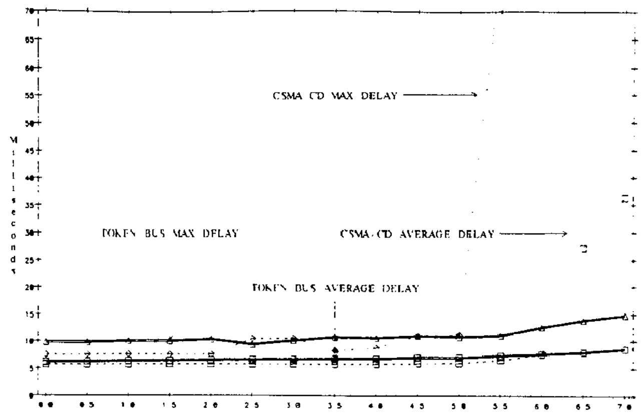

Figure 7. Transport User Average and Maximum One-way Delay for 20 Millisecond Status Reporting

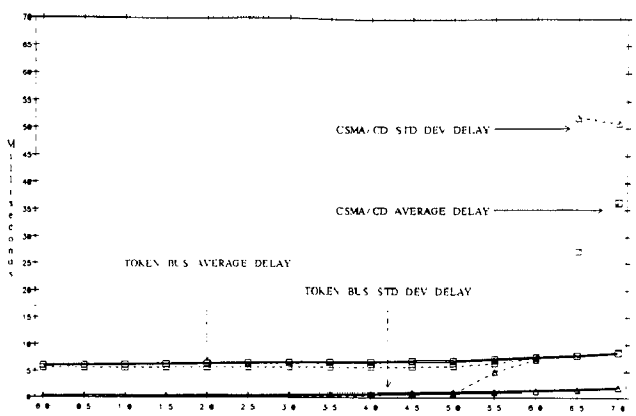

Figure 8. Transport User Average and Standard Deviation One-way Delay for 20 Millisecond Status Reporting

If the application is sensitive to variance in one-way delays, Figure 8 shows that the advantage of CSMA/CD over token passing is reduced. The standard deviation in one-way delays is small when operating over both types of networks, up to a load of 50%, after which the deviation in one-way delays for the CSMA/CD network increases rapidly. Therefore, if loads of 50% or less are expected there is no clear advantage for either access method because the standard deviation in one-way delays is below 1 ms in all cases.

When the application must provide a guaranteed maximum one-way delay, Figure 7 illustrates that token passing has a potential advantage over CSMA/CD as the background load passes 50%; but, Figure 7 does not demonstrate the uncertainty associated with the CSMA/CD access method. An individual message can be substantially delayed due to repeated collisions. Therefore, one can expect significantly higher maximum delays with CSMA/CD at loads above 40%. Token passing provides a more controlled bound on the maximum one-way delay.

V. Request-Response Performance Predictions

Many factory applications entail requests for information and an associated response. The experiments detailed below examine the user reponse time obtained for such applications. In the first experiment, requests arrive at a constant rate without regard to previous responses. In the second experiment, a request is issued each time the response is received for a previous request. Figure 9 illustrates the method used to measure response time. The requesting user issues a 20 byte TSDU (i.e., the request) at time Treq, the TSDU is encapsulated as a TPDU and sent across the network to a responder. Upon receiving a request, the responding user submits a 200 byte TSDU (i.e., the response), the TSDU is formatted as a TPDU and sent across the network, arriving at the

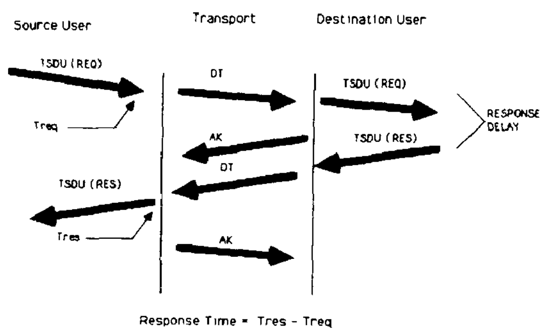

FIGURE 9 Measurement of Response Time

requester at time Tres. The reponse time is measured (Tres-Treq) and accumulated for later calculation of the mean and other distribution statistics. The experiments are conducted over both CSMA/CD and token bus LANs with background traffic varying from 0 to 70%. The composition of the background traffic is 200 byte messages generated by a Poisson arrival process.

A. Constant Rate

The constant rate experiment measures response times between a requesting and responding user over a single transport connection. The requester issues a new message every 20 ms. The responder issues a response immediately upon receipt of a request. The results obtained are given in Figures 10 and 11.

For loads of 55% and below, the user sees superior average response times using CSMA/CD. If the application is more sensitive to variance in response time, CSMA/CD still provides superior performance but only for loads of 45% and below. Even for maximum response time the advantage of CSMA/CD is demonstrated for loads of 40% and below.

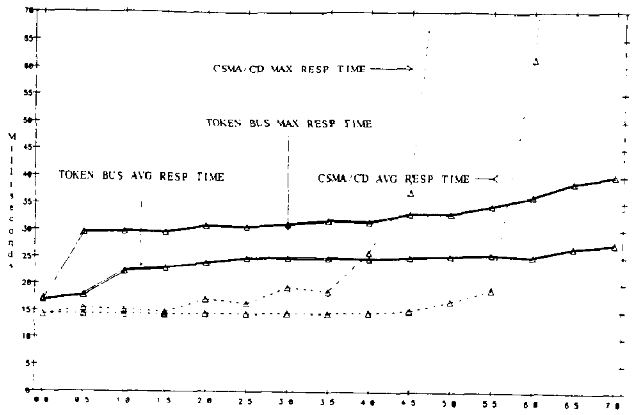

Figure 10. Transport User Average and Maximum Response Time for Constant Rate Request-Response

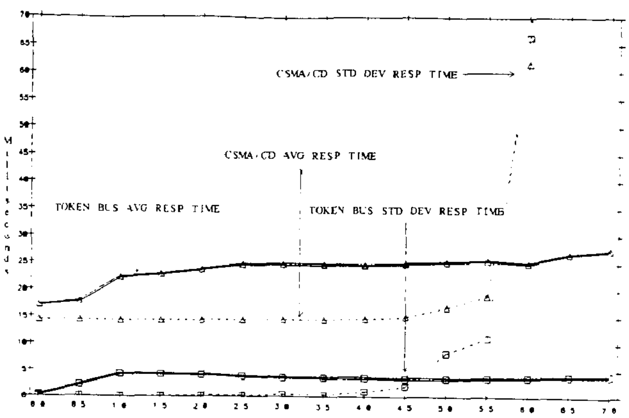

Figure 11. Transport User Average and Standard Deviation Response Time for Constant Rate Request-Response

B. Maximum Rate

For the maximum rate experiment the requesting user submits requests as fast as responses are returned, thus, the message arrival rate adapts to the network load. The results are given in Figures 12 and 13. At high loads, the average, standard deviation, and maximum response times are smaller than the same measures made with a constant arrival rate. The users only load the transport stations at the sustainable rate and no significant queuing occurs within the transport. In the previous experiment, as the background load increased, the users continued to load the transport station at a constant, non-sustainable rate and significant transport queuing occurred.

Interestingly, at low loads, the average, maximum, and standard deviation of response times for the maximum rate experiment are sometimes higher than for the previous constant rate experiment. The explanation is simple. At low loads, where the average response time is below 18 ms, a dead time of 2 ms or more exists before the next request arrives. During this dead time, the two transport stations perform the processing associated with the AK TPDU for the response DT TPDU. When the maximum rate arrival scheme is used, no significant dead time

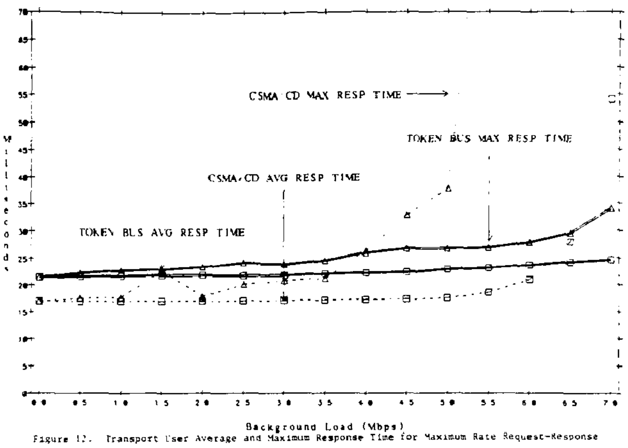

Figure 12. Transport User Average and Maximum Response Time for Maximum Rate Request-Response

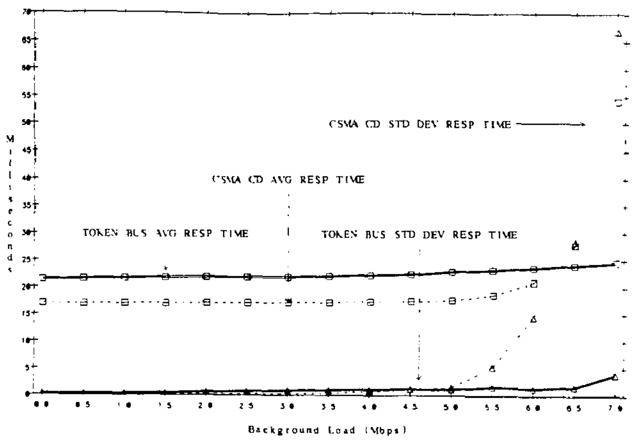

Figure 13. Transport User Average and Standard Deviation Response Time for Maximum Rate Request-Response

exists between user receipt of a response and generation of the next request, thus the request must queue behind the AK TPDU for the previous response. This queuing occurs at both transports and a total of 2.7 ms is added to the one-way delay of the request message.

With maximum rate arrivals, the user sees better average response times over a CSMA/CD network for loads of 60% and below, but CSMA/CD provides no advantage over token passing when the measures of importance are standard deviation and maximum response times For loads of 40% and below, CSMA/CD yields equal or better response time performance than token passing for maximum rate request-response applications.

VI. Conclusions

This paper described a detailed, five-layer simulation model of OSI protocols and demonstrated use of the model to predict transport protocol performance. The findings, although not yet validated, suggest that OSI protocols, implemented with currently available technology and used on small, factory floor networks, can provide typical user throughputs of 1.5 Mbps, one-way delays between 6 and 10 ms, and response times between 15 and 25 ms. These results, restricted to single connections with the traffic patterns as specified, are sufficient to encourage continued work to validate the simulation model through live experiments. The model is now being used at the NBS to plan experiments exploring multi-connection applications with varied user traffic.

With respect to choice of network access method, the results indicate that for network loads of 40% and below, CSMA/CD provides performance equal to or better than token passing bus. This is shown for the average, standard deviation, and maximum response times and one-way delays.

For one-way file transfers, token passing bus permits greater throughputs than CSMA/CD. At low loads, the transport stations cannot take full advantage of the low access delays of CSMA/CD because of an imbalance in CPU requirements between sender and receiver. As the load increases, the increasing access delay of CSMA/CD pushes transport throughput even lower. At very high loads MAC layer discard of packets forces the transport layer retransmission procedures into action, lowering transport throughput even further. For two-way file transfers, the pattern seen is much the same, although the transport stations can take full advantage of the low access delays of CSMA/CD at low loads to achieve throughput superior to that available from token passing.

For small, flexible manufacturing networks, where loads below 40% can be expected the CSMA/CD access method can provide better performance for time-critical applications than token passing bus. When loads between 40% and 70% are anticipated token passing bus will yield superior time-critical performance.

REFERENCES

ARC84 J-L Archambault, An IEEE 802.4 Token Bus Network Simulation, NBSIR 84-2966, National Bureau of Standards, October 1984.

COL85A R. Colella, R. Aronoff, and K. Mills, "Performance Improvements for ISO Transport," Proceedings: Ninth Data Communications Symposium, September 1985.

COL85B R. Colella, M. Wheatley, and K. Mills, COMSAT/NBS Experiment Plan for Transport Protocol, NBSIR 85-3141, National Bureau of Standards, March 1985.

IEE84A IEEE 802.2 Standard for Logical Link Control, December 1984.

IEE84B IEEE 802.3 Standard for Carrier Sense Multiple Access with Collision Detection, December 1984.

IEE85 IEEE 802.4 Standard for Token-Passing Bus Access Method, February 1985.

INT84 iSBC 186/51 COMMputer Board Hardware Reference Manual, Order Number 122136-002, Intel Corp., 1984.

HAB85 L. Haber, "Cooperative MAP Standards Effort Nets Result," Mini-Micro Systems, November 1985, pp. 50-54.

HEA85A S. Heatley and R. Colella, Experiment Plan: ISO Transport Over An IEEE 802.3 Local Area Network, (under preparation).

HEA85B S. Heatley, "A Simplified Model for an IEEE 802.3 LAN," (under preparation).

MAP85 Maufacturing Automation Protocol, Version 2.1, General Motors, March 1985.

MCL85 J. McLeod, "Autonomous Networks Begin to Communicate," Systems & Software, November 1985, pp. 44-58.

MIL85 K. Mills, M. Wheatley, and R. Colella, "Simulation of an International Standard Transport Protocol," 1985 CMG XVI Proceedings, December 1985.

NRC85 FUSION Network Software Version 2.0, Network Research Corporation, Los Angeles, California.

OSI82 Information Processing Systems - Open Systems Interconnection - Basic Reference Model, IS7498, International Organization for Standardization, 1982.

RYD85 M. Ryder, J. Sagli, and K. Won, "Module puts Factory Devices on the MAP," Systems & Software, September 1985, pp 77-79.

SHO80 John F. Shoch and Jon a Hupp, "Measured Performance of an Ethernet Local Network," Communications of the ACM, December 1980, pp. 711-721.

TOE83 Robert E. Toense, "Performance Analysis of NBSNET, Journal of Telecommunications Networks, Volume 2, Number 2, 1983, pp. 177186.

TOP85 Technical and Office Protocols Specification Document, Working Draft Version X.4.0, Boeing Computer Services, September 1985.